Analysis of SQL Injection based Attack Vectors on Elitepvpers.com

|  |
| --- |
| Corey Hoffman  November 27, 2014  CS1699 Student  Cah150@pitt.edu |

# Abstract

This paper details the process and results of a SQL Injection security analysis test. It describes the methodology taken and the reasoning behind the conclusion that elitepvpers.com is sufficiently resistant to SQL based attacks.

# Background

In order to understand some of the reasoning and language used throughout this paper some background is necessary. Below please find information on topics relating to the subject material.

## Elitepvpers.com

Elitepvpers.com is a forum and trade hub for gamers. It is renowned for its “blackmarket” providing a center for trading of in-game resources and services. Due to the occasionally illicit nature of the activities performed the forum is often subject to attack and likely secure as a result.

## SQL Injection

SQL Injection is a hacking technique that attempts to exploit raw inline query strings. By inputting malicious SQL into a field that in-turn directly inserts that input into a SQL query, this allows the malicious entity the potential to alter the initial purpose of the hard coded statement an exert control over the underlying database.

## Robots.txt File

The robots.txt file is a plain text document stored in the root directory of a website. The purpose of a robots.txt file is to specify to web crawlers which pages they may or may not visit when collecting data about the website. While it is easy for illegitimate web crawlers to avoid this document, legitimate ones such as those belonging to Google will respect its contents. It is a great document to look at while attempting to find security vulnerabilities.

## Misc. Terms

Terms used in the context of this paper:

* Attack Vector – Any method by which a malicious user can “Attack” or attempt to gain control over a program or system.
* Vulnerability – Any fault in design that allows a malicious user access, e.g. a weakness.

# Approach

In order to properly assess the security of the website under test I first needed to identify potential attack vectors. While there are plenty of automated tools available online to accomplish this task I decided, for the sake of this project, to manually evaluate the security of this website.

SQL Injection susceptible targets generally include fields on web forms or URL based query strings used as input to .php scripts. With this knowledge I identified several web form targets immediately (some were behind a login wall so I made an account). These were the username and password fields, the password change field and some fields under the profile settings that dealt with personal information and such. I proceeded to apply some standard tests such as ending the input data in a ‘ or “ in order to see if there was any response. There was no response; the inputs to the fields were all parameterized as expected.

This is a good time for a disclaimer, Since I did not own the website under test all attempts to find attack vectors were carried out in a way that would only read publically accessible data and not modify anything or reveal any private user data in the event of a discovered weakness.

My next step was to try and identify php scripts that may not have parameterized inputs, in my opinion this is far more likely that web form fields to be a successful attack vector. In order to find likely candidates I decided to explore the contents of the websites’ robots.txt file.

Using the robots.txt file I identified two promising .php files, these were the “search.php” and “post\_thanks.php” files. I soon discovered that “search.php” handled all queries made regarding the contents of the websites forum and “post\_thanks.php” handled queries related to searching and giving “thanks” on the forum. Presumably the “post\_thanks.php” module was a feature added after the “search.php” functionality was implemented.

Having identified these two potential vectors of attack I proceeded to find instances of their usage on the website. Once I had found such cases I attempted a manual injection into the URL based query string of each script hoping that those would reach the final query raw. They were well handled and did not provide any useful information. Having exhausted obvious attack vectors I deemed my task complete.

# Results

It is my opinion that the website Elitepvpers.com is well fortified against SQL Injection based attacks. This opinion is supported by the test data. I tested each vulnerable field on all web forms likely to be susceptible to an SQL Injection attack. Additionally, all query string input to potentially insecure .php pages.

There are caveats to these results, as stated in my approach I attempted to test only fields that returned data by query and I did not attempt any write based attacks, this is because I did not own the website under test and I did not wish to unintentionally perform any illegal activities in the event that I did uncover a vulnerability. It should be noted that since all of the tested fields were parameterized or parsed it is unlikely to have made a difference.

# Conclusion

After a sufficient amount of security testing, I have reached the conclusion the elitepvpers.com is sufficiently resistant to SQL injection based attacks. This conclusion comes with the caveat that these tests were not performed by an automated system and is therefore not exhaustive in coverage. Additionally, these tests were not intended to cause harm in anyway and were therefore performed carefully with read only based injections.